

TDDD80 Mobila och sociala applikationer

Kursintroduktion

Personal

- Rita Kovordanyi <rita.kovordanyi@liu.se>
 - Kursledare, föreläsare, seminarieledare
- Anders Fröberg <anders.froberg@liu.se>
 - Kursledare, föreläsare, seminarieledare
- Veronica Gunnarsson
 - Kursadministratör

Kursassistenter

- Simon Bonnedahl,
- Jody Foo, jody.foo@liu.se
- Sahand Sadjadee, sahand.sadjadee@liu.se
- Josef Atoui, josat799@student.liu.se
- Chu Wanjun, chu.wanjun@liu.se
- Syed Haider, syed.muhammad.arslan.haider@liu.se

Kursförbättringsarbete

- Förra årets utvärdering
 - Bra överlag
 - Vill ha *en* lösning på Android-labbar (inte välja mellan gamla och nya sättet)
- Vi kör nya sättet fullt ut på Android
- Lärarteam-möten varje vecka där vi gå igenom veckans labbar, föreläsningar, och live-kodning, så att allt är samstämmigt



Lärandemål enligt kursplan

Designa, implementera och utvärdera:

1. Webbaserade mjukvarutjänster
2. Användbara och vältestade mobila applikationer med sociala element
3. Tillämpa och reflektera kring relevant metodik och praktik för individuell programutveckling

Lärandemål 1

- Designa, implementera och utvärdera webbaserade mjukvarutjänster
 - Bygga en central databas
 - Tillhörande mjukvara (server) som lägger in och hämtar data ur databasen
 - Server: gör *tjänster* åt den som anropar (t.ex. mobilapp)
 - T.ex. tillhandahåller eller sparar data

Lärandemål 2

- Designa, implementera och utvärdera användbara och vältestade mobila applikationer med sociala element
 - Social app i Android
 - Projekt som spänner över alla faser i mjukvaruutveckling
 - Idé och design
 - Tidsplanering
 - Informationssökning
 - Implementering
 - Testing och utvärdering

Lärandemål 3

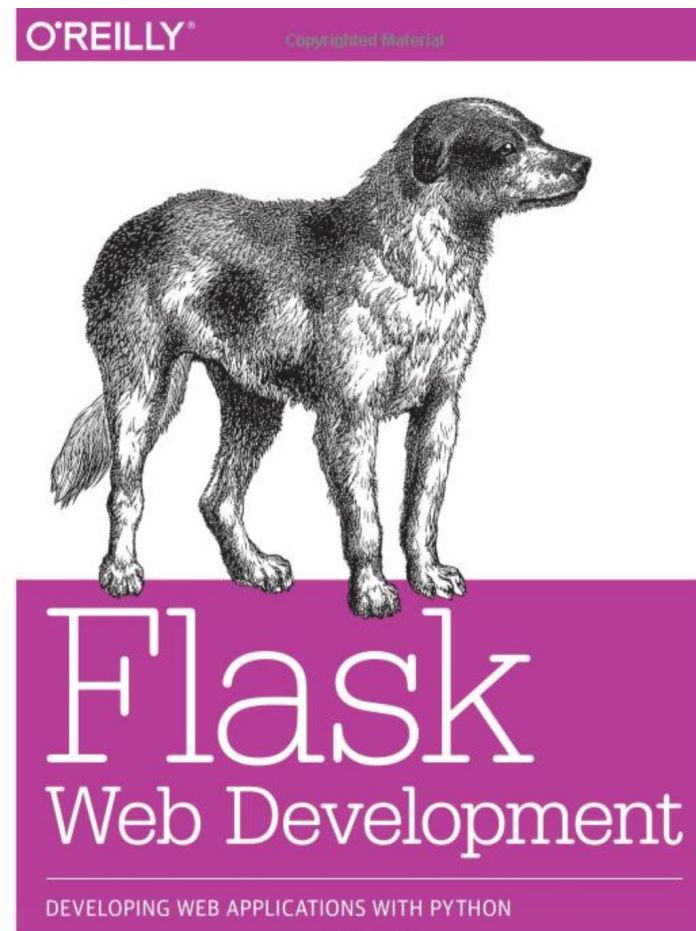
- Tillämpa och reflektera kring relevant metodik och praktik för individuell programutveckling
 - Versionshantering, kodgranskning, testing, underhållbarhet
 - *Utveckla en förmåga till egen kompetensutveckling*
 - *Viktig praktik: Att söka information*

Olika sätt att ta reda på...

- Teoretiska begrepp, kodstil, etc. som du vill veta mer om

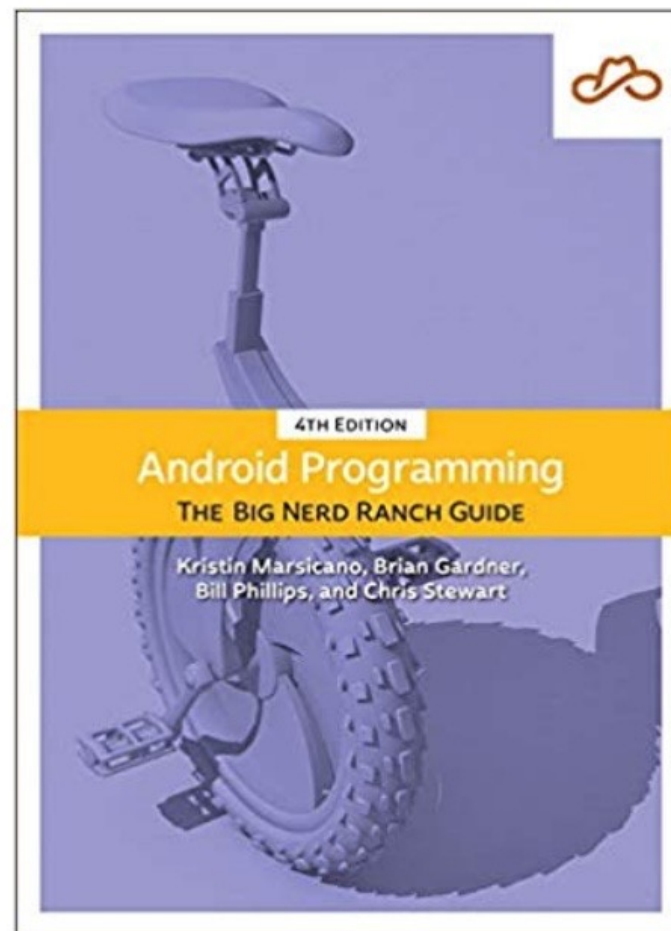
Referenslitteratur

Flask Web Development:
Developing Web Applications
with Python



Referenslitteratur

Android Programming:
The Big Nerd Ranch Guide



Bra online alternativ till pappersbok


Safari File Edit View History Bookmarks Develop Window Help

https://exploreflask.com/en/latest/

book to learn A... ViewModel | A... best book flask... are best resour... The Flask Meg... place to learn... I want to learn... book about Fla... Explore Flask... Python3 Tutori... to g...

Search docs

Preface
Coding conventions
Environment
Organizing your project
Configuration
Advanced patterns for views and routing
Blueprints
Templates
Static files
Storing data
Handling forms
Patterns for handling users
Deployment
Conclusion


3 days, 6 keynotes, 400 attendees, 30 speakers: PyCon Colombia 2018, Feb 9-11
Ads served ethically

Read the Docs v: latest

- [A quick primer on Jinja](#)
- [How to organize templates](#)
- [Inheritance](#)
- [Creating macros](#)
- [Custom filters](#)
- [Summary](#)
- [Static files](#)
 - [Organizing your static files](#)
 - [Manage static assets with Flask-Assets](#)
 - [Summary](#)
- [Storing data](#)
 - [SQLAlchemy](#)
 - [Summary](#)
- [Handling forms](#)
 - [Flask-WTF](#)
 - [Summary](#)
- [Patterns for handling users](#)
 - [Email confirmation](#)
 - [Storing passwords](#)
 - [Authentication](#)
 - [Forgot your password](#)
 - [Summary](#)
- [Deployment](#)
 - [The Host](#)
 - [The stack](#)
 - [Summary](#)
- [Conclusion](#)

Thank you

First of all, I'd like to say thank you to my volunteer editor, Will Kahn-Greene. He was great about going over my very rough drafts and helping me decide on the scope and direction of the content. I'm looking forward to working with him to manage the project into the future.



Platform

<https://developer.android.com/docs/>

earch

OVERVIEW

GUIDES

REFERENCE

SAMPLES

DESIGN & QUALITY

Documentation for app developers

Whether you're building for Android handsets, Wear OS by Google, Android TV, Android Auto, or Android Things, this section provides the guides and API reference you need.

Get started

[Build your first app](#)[Sample code](#)[API reference](#)[Design guidelines](#)[Codelab tutorials](#)[Training courses](#)

Android devices

[Wear OS](#)[Android TV](#)[Android Auto](#)[Android Things](#)[Chrome OS Devices](#)

Best practices

[Testing](#)[Performance](#)[Accessibility](#)[Security](#)[Enterprise](#)[Emerging markets](#)

Apputveckling idag

- Ständigt förändrade förutsättningar
 - Bättre telefoner, snabbare nätverk
 - Språk och 'best practices' uppdateras
 - Mjukvaruverktyg och metodologi förändras
- Pappersböcker blir snabbt föråldrade
- Bättre att utnyttja resurser på nätet

Även expert-programmerare Googlar

“... programmers: they're not walking encyclopedias...”

<https://www.quora.com/Do-expert-programmers-google-frequently-when-coding>

“There's a difference between googling to understand the problem and just finding some code to copy/paste”

<http://programmers.stackexchange.com/questions/114002/is-it-a-good-idea-to-always-use-google-as-the-first-step-to-solving-a-problem>

Sökning, ex 1 (snabbfix)

- Jag har kört fast (får t.ex. körningsfel eller syntaxfel), och förstår inte varför
- Oftast har någon annan undrat samma sak
 - Kolla på forum (skriv in felmeddelandet i sökrutan)
 - Stack Overflow
- Snabbsökning, räcker oftast med ett par källor

Sökning, ex 2 (djupsökning)

Jag känner att jag inte förstår begrepp X:

1. Leta i dokumentation, t.ex. 'Android developer pages'
2. Tutorials (Youtube, etc.) – t.ex. Google I/O talks
3. Tutorials (av andra än utvecklarna själva)
4. Leta fritt (t.ex. forum som Stack Overflow)

Läs flera källor tills du känner att du har förstått

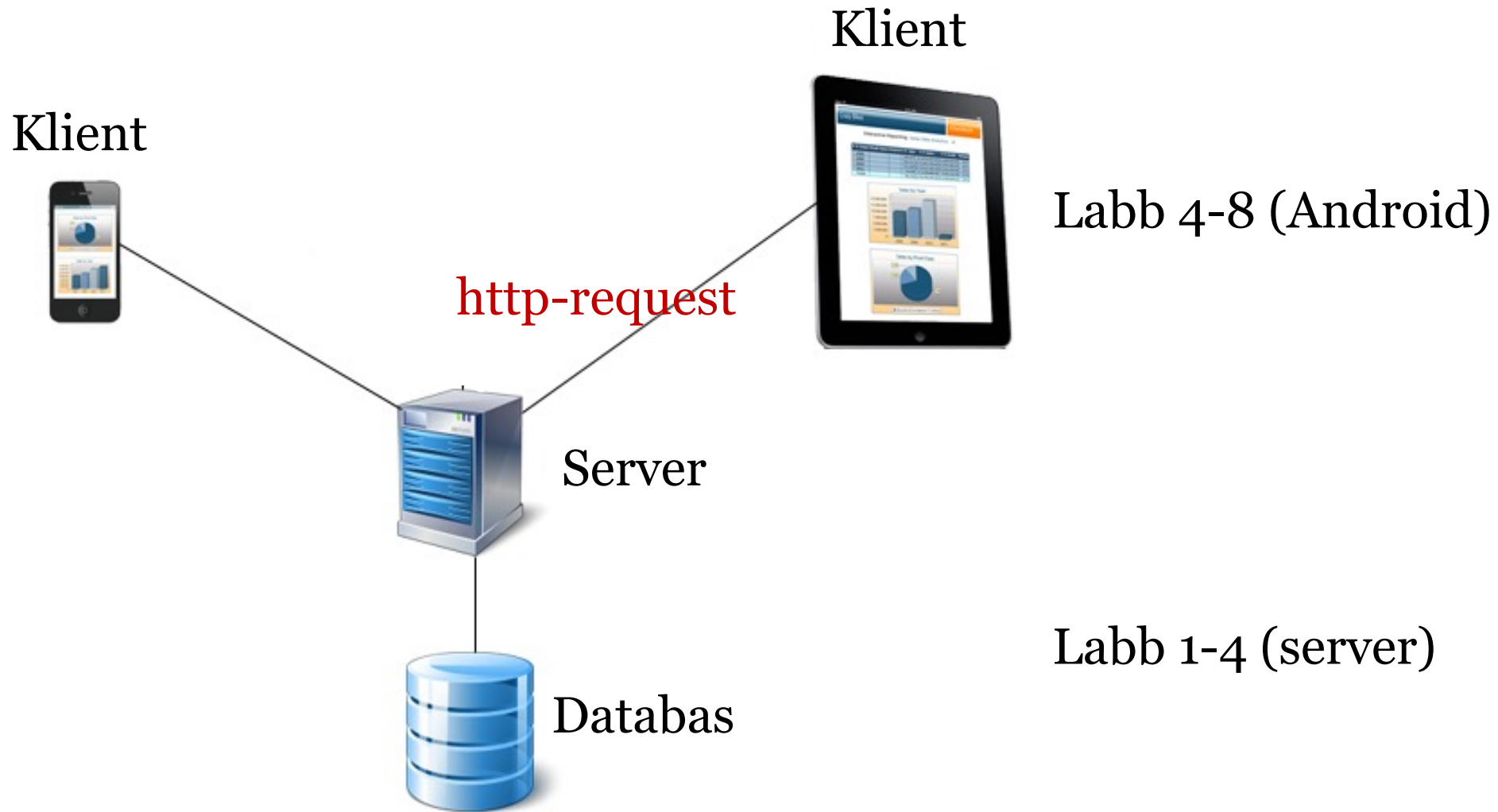
Informationssökning i kursen

- Lätt att man söker efter snabba lösningar
 - Kod att kopiera...
- Betoning på *sökning för kunskapsinhämtning*
 - Krav i kursen!
- Varje labb börjar med *handledd informationssökningspass*
 - Leta svar på ett antal begreppsfrågor
 - Redovisas i README-fil, med övrig labb-kod i Gitlab

Kursupplägg

Labbar varvat med projektarbete

- 2 veckor labb
 - Tillämpa i egna projektet (1 vecka)
- 2 veckor labb
 - Tillämpa i egna projektet (1 vecka)
- Etc.



Fusk-klient

http-request



Server



Databas

Labb 1-4 (server)

Http-request (exempel)

POST /resource/1 HTTP/1.1

Host: example.com/message/

Content-Type: application/json; charset=UTF-8

```
{  
  'name': 'Pelle',  
  'message': 'Hej, hur är läget?',  
  'device': 'dev001'  
}
```

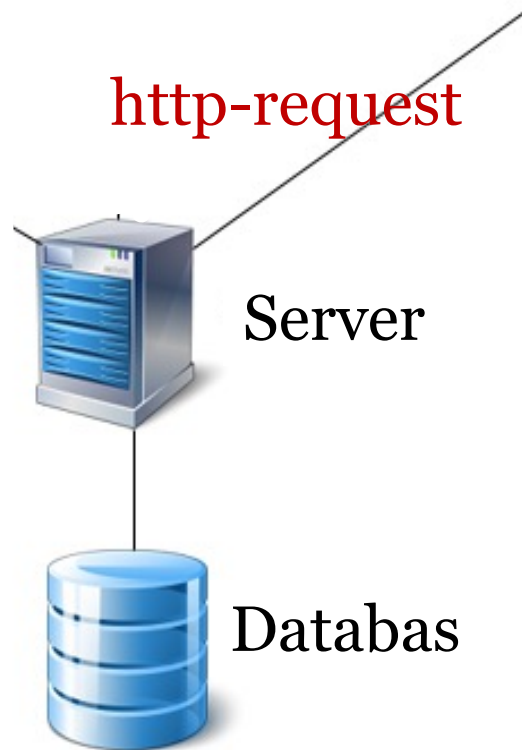

Fusk-klient skickar http-request till servern

- Anrop till servern
 - Vill antingen hämta information från servern (GET)
 - Eller lägga till information (POST)
 - Eller ta bort (DELETE)

I python:

Fusk-klient

```
requests.post('http://example.com/', json={'message': 'Hej'})
```



Labb 1-4 (server)

Tidslinje

- Labbar
- Appidé-workshop
- Labbar varvas med projektarbete (vt1)
 - Koda de delar av egna appen som ni just lärt er på labbarna
- Seminarier
 - Kodgranskning
 - Underhållbarhet
- Användbarhetsutvärdering av er app

Labbarna

Labb och projektarbete

- Genomförs i par
- Notera: inte alla tillfällen handledda
 - Vissa tillfällen har vi reserverat platser, så att ni har tillgång till datorer, men ni förväntas arbeta själva

Labbar

- 1 labb / vecka
 - Ca. tre tillfällen / labb
- Börjar med handledd informationssökning
 - Ett antal frågor och begrepp som ni ska reda ut genom att söka på nätet
 - Ger bättre förståelse för labben och de lösningar som ska tillämpas senare i projektet

Elektroniskt kösystem under labbarna

- Vi använder web-appen Camedin
 - Ställa sig i kö för handledning under labben
 - Ni skriver upp er med ert datornummer
 - En kö för varje labbhandledare
 - Ni kommer att få lösenord av er handledare
- Camedin tutorial
 - <https://www.youtube.com/watch?v=k06p2djJJis>





Camedin inloggning

Session: 1 2016-01-13 10:15:00 - 2016-01-13 15:00:00




Course Name:
Web Programming TDDD97

Study Period:
2016-01-11 - 2016-03-08

Assistants:

-  Sahand Sadjadee
-  Filip Marko
-  Gustav Bylund
-  Alexander Kazen

Available Assistants

Assistant	Queue	Place
 Sahand	0	-
 Filip Marko	0	-
 Gustav Bylund	0	-

Refresh Page

Your full name*

Your location

Password*


Assistant*

- Sahand Sadjadee - 0
- Filip Marko - 0
- Gustav Bylund - 0
- Alexander Kazen - 0

Information About This Session

Kurs: TDDD97, Laboration, SU00, SU01, SU02, SU03, SU04, C3, CS1, DAV1, DI3

Lab Sessions



Total: 19
Left: 19

Labbredovisning

- På fredagar: Dema er lösning för *er* handledare
- Var beredda att *individuellt* besvara frågor från labbhandledaren
- Lämna in kod i Gitlab (<https://gitlab.liu.se>)
 - Lämna in svar på informationssökningsfrågorna i en README-fil tillsammans med koden i Gitlab
 - Bjud in din labbhandledare som ‘reporter’ till repot i Gitlab

Projektet

Genomförs i par

Projektet

- Ett socialt fleranvändarsystem för Android designas, utvecklas, testas, och utvärderas
- Systemets syfte, funktion och innehåll är valfritt, givet att betygskriterierna uppfylls
- Systemet måste ha en specifik målgrupp och lösa ett äkta problem på ett professionellt sätt
- Avsikten ska vara att kunna lansera systemet efter kursen



It's cold!

16/01/2015

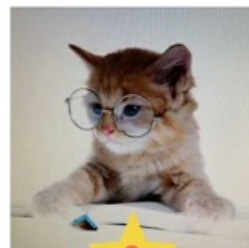


Rate Now!

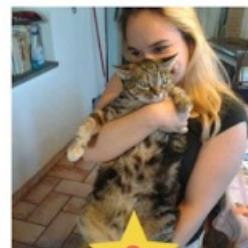
Next Cat >



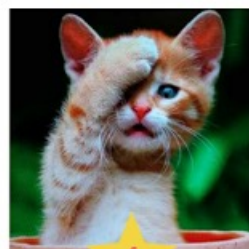
Lookatmeeee
Average: 5



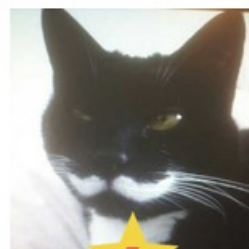
Schrodinger
Average: 5



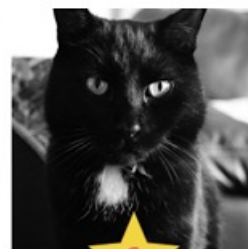
fattyfatty
Average: 5



Mr Potter
Average: 4.7



Sir Purrsalot
Average: 4.3



Stig
Average: 4.3



Buttercup



Massive



DJ Whiskers



Vad får man göra på Guldbaggegalan?

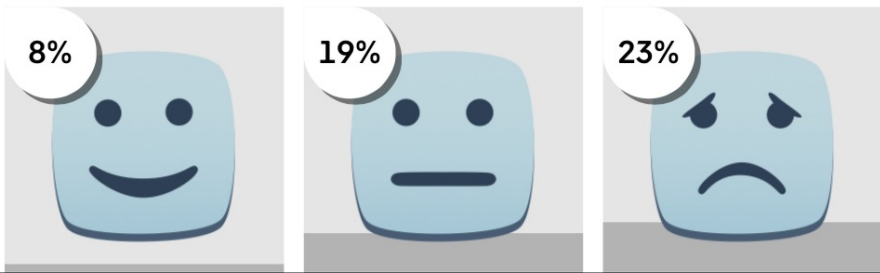
Årets Guldbaggegalan har rört upp känslor. Att "göra en Gösta" har blivit ett begrepp. Hur får man egentligen uppföra sig på en gala?

Gäster är Suzanne Osten, regissör, och Kristoffer Appelquist, komiker.

Hur känner du?

22:32

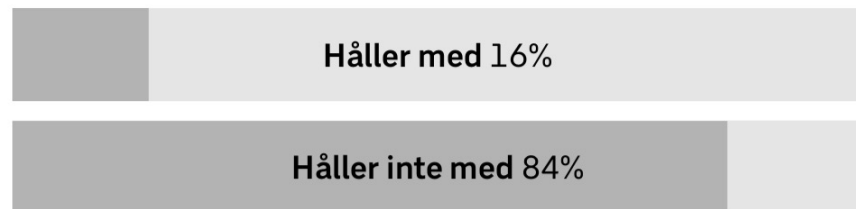
Hela diskussionen om Kommunal: Hur känner du?



Håller du med?

22:42

"Guldbaggevinnarna borde vara ännu mer politiska i sina tacktal." Håller du med?

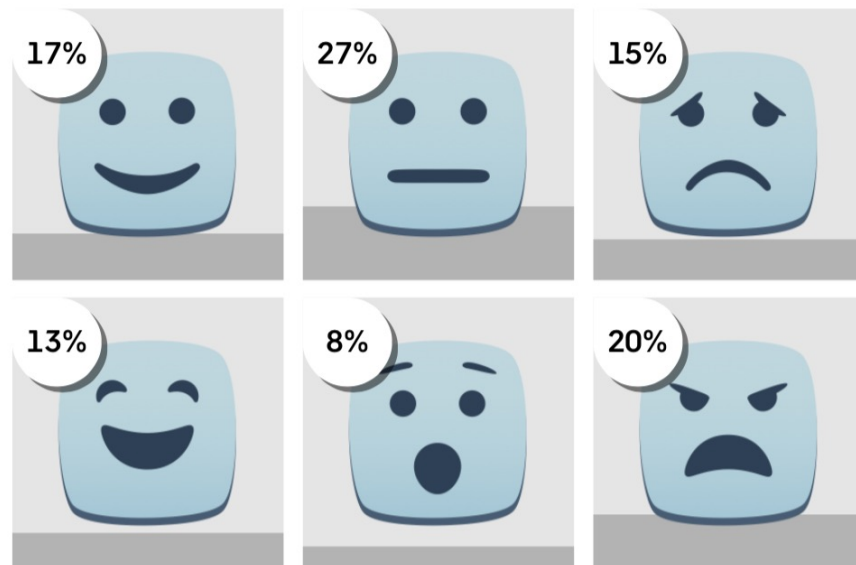


Röster: 198

Hur känner du?

22:40

Samtalet om Guldbaggegalan: Hur känner du?



Röster: 209

Exempel på poänggivande krav

- Sociala funktioner
 - Gilla informationsobjekt (1p)
 - Kommentera informationsobjekt (1p)
 - Följa andra användare (1p)
- Användning av kamera
- Användning av kamera
- ...

Kom gärna med egna förslag på features som kan vara värda poäng

Exempel på poänggivande krav

- Tidsplaneringförmåga
 - Lämnat in server-labbar i tid (1 p)
 - Lämnat in Android-labbar i tid (1 p)

Projektbetyg

- Betyg 3: minst 4p
- Betyg 4: minst 6p
- Betyg 5: minst 8p

Tidslinje för projektet

Projektet

1. App-idé workshop VT1
 2. Databas-del (backend)
 3. Kodgranskningssem. → **Deadline för överl. kod: 21/2**
 4. Första skärm (Android)
-

- VT2
5. Fortsatt implementering
 6. Underhållbarhetssem. → **Deadline för överl. kod: 12/4**
 7. Användbarhetstestning
 8. Slutdemo + kodinlämning
-

Seminarier

VT1

1. Kodgranskning (server-kod)
 - Granska annan students kod
 - Kodstil, kodkvalitet
-

VT2

2. Underhållbarhet
 - Koda föratt underlätta underhåll av mjukvara

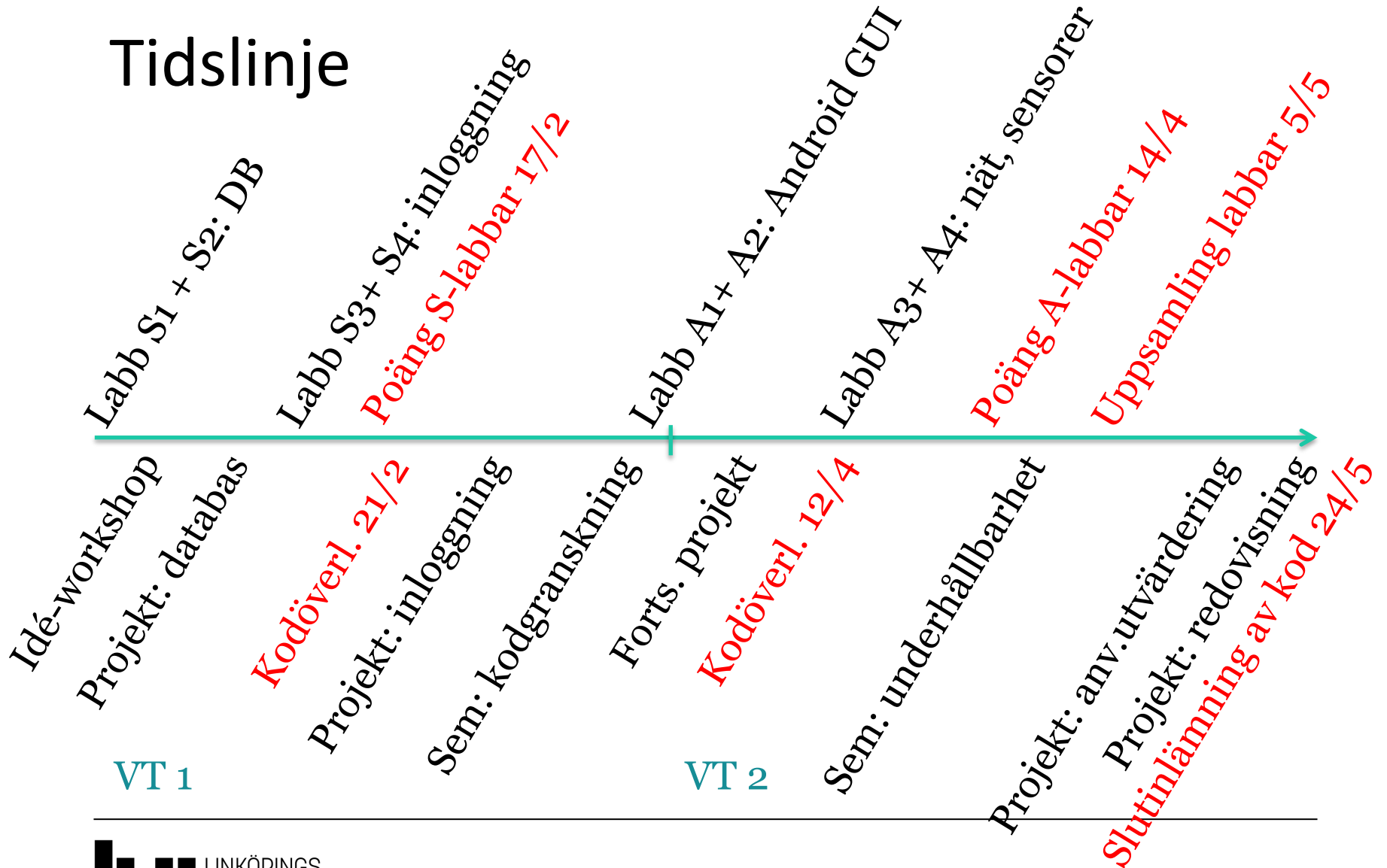
Seminarierna

- Viktigt att ni har skrivit upp er i sem/projekt-delen i webreg **före kodinlämningen**
 - Två grupper:
 - Sem grupp A = webreg grupp 1, 2, 3
 - Sem grupp B = webreg grupp 4, 5
 - Egen kod skickas till det studentpar som kommer efter i listan i webreg
 - Sista (t.ex. sista i gr 3) skickar till första på listan (första i gr 1)

Projektredovisning

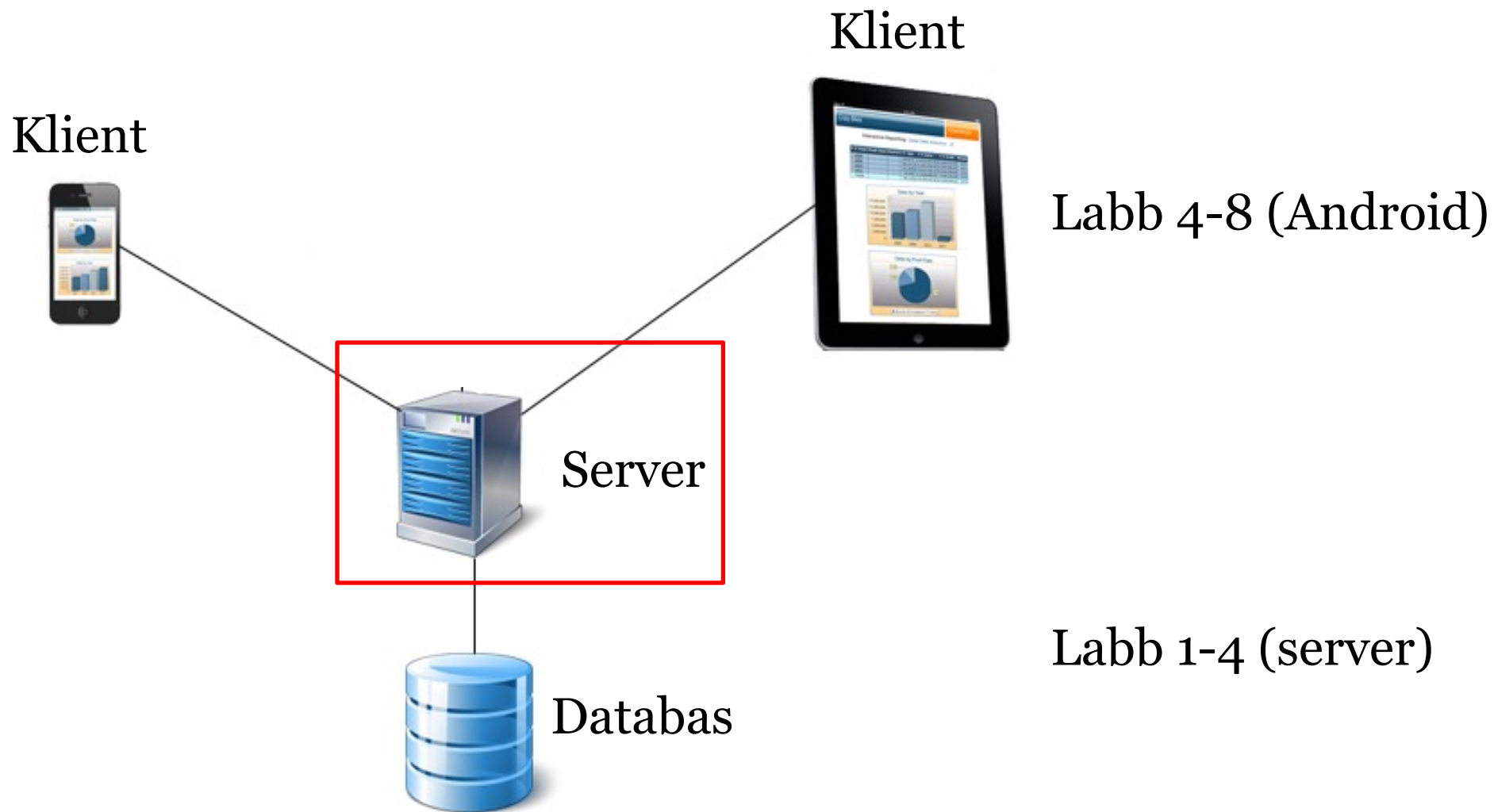
- Muntlig demo
 - Tänk på att test coverage ska demas och dokumenteras
- Rapport om användbarhetsutvärdering
 - Lämnas in i gitlab

Tidslinje



Server-kod

TDDD80 Mobila och sociala applikationer



http-request skickas från klient till server

POST /resource/1 HTTP/1.1

Host: example.com/login/

Content-Type: application/json; charset=UTF-8

{

'email': user_email,

'password': user_pw

}

header

body

Serverns svar

HTTP/1.1 **200 OK**

Cache-Control: no-store

Pragma: no-cache

...

Vanliga responskoder

- 200 OK
- 300 Redirect till annan adress
- 400 Fel från sändaren (fel adress, fel metod)
- 500 Fel i servern (bug, ej implementerad funktion)

HTTP

HTTP – HyperText Transfer Protocol

- Kommunikationsprotokoll
 - Kan användas för att efterfråga och skicka web-sidor
 - Kan även användas till att skicka data
 - T.ex. formaterade text-strängar
 - Ex. {‘bok-titel’: ‘Krig och fred’,
‘pris’: 195
‘kapitel’: [...]}
}

Göra anrop under de första labbarna

- Vi har inte klienten ännu, utan kommer att anropa vår server-kod från terminalfönster
 - Använd gärna biblioteket requests
 - Kan rada upp requests och bearbeta responses i en sammanhållen .py fil
 - Kör .py-skriptet för att göra en serie http-anrop
 - Snabbar upp arbetet med labbarna

Anrop (request) till server

- Från script (fake klient)

```
URL_root = 'http://localhost:8080'
```

```
response = requests.get(URL_root + '/lunch_menu/')
```

- Från klienter (Android)

```
HttpGet req = new HttpGet("../lunch_menu");
```

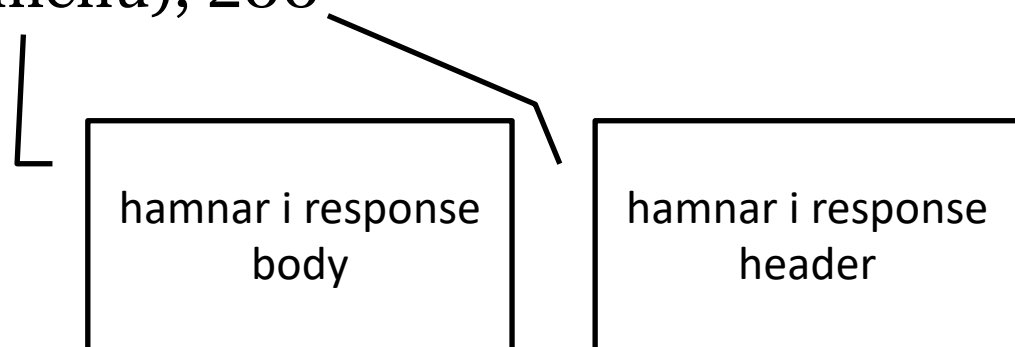
```
response = client.execute(req);
```

Flask

På server-sidan

- Tas emot av server-kod (skriven i Flask)

```
@app.route('/lunch_menu', methods=['GET'])  
def fetch_menu():  
    menu = database.find_todays_menu()  
    return jsonify(menu), 200
```



Olika respons-koder

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

- 2xx – success
- 400 – ‘bad request’ (syntaxfel på anropet)
- 404 – fel URL / ingen sån resurs
- 405 – ingen metod angivet i anropet / matchar inte tillåtna metoder för denna route i server-koden

Vanliga request methods

- http://www.w3schools.com/tags/ref_httpmethods.asp
 - GET—Fetch data
 - POST—Ask server to upload enclosed data
 - PUT—Save or update enclosed document
 - DELETE—Delete the specified resource

JSON

JSON (JavaScript Object Notation)

```
{ "menu":  
  { "id": "file",  
    "value": "File",  
    "popup": {  
      "menuitem": [  
        { "value": "New", "onclick": "CreateNewDoc()" },  
        { "value": "Open", "onclick": "OpenDoc()" },  
        { "value": "Close", "onclick": "CloseDoc()" }  
      ]  
    }  
  }  
}
```

<http://json.org/example.html>

Till/från JSON

- Server-sidan

Flask jsonify(**dict), se t.ex.

<http://stackoverflow.com/questions/13081532/how-to-return-json-using-flask-web-framework>

- Klient-sidan (Android/java) är gson ett populärt bibliotek

```
Person p = gson.fromJson(reader, Person.class);
```

Utvecklingsmiljön

TDDD80 Mobila och sociala applikationer

Utvecklingsmiljö

- PyCharm (senare Android Studio)
- Flask
 - Python-baserat “språk” för server-kod
- Requests-biblioteket

Virtual environment

- Sätt att låta olika projekt använda olika python-versioner, och olika bibliotek, utan att störa andra projekt
- Läs mer på <https://docs.python.org/3/tutorial/venv.html>

Flask

- Quickstart Flask:
 - <http://flask.pocoo.org/docs/quickstart/#quickstart>
- Tutorial för Flask:
 - <https://flask.palletsprojects.com/en/2.0.x/tutorial/>

www.liu.se